

哈爾濱工業大學

# 网络安全实验报告

题    目 基于 libnet 的程序设计

专    业 信息安全

学    号 1170301027

学    生 冯帅

指导教师 王彦

## 一、实验目的

掌握 libnet 数据包的构造原理。

## 二、实验内容

1. 掌握 libnet 数据包的构造原理
2. 编程实现基于 libnet 的数据包构造，结合前面实验给出验证过程。能够对源码进行解释。

## 三、实验过程

基于 libnet 的数据包构造实验基本信息：

实验环境：Ubuntu18.04 x64

编程语言：C 语言

### 1. 需求分析

需要使用 libnet 构造并发送一个数据包，并验证这个数据包被成功发送了。验证这一过程需要用到实验二中的捕包程序 capture，将生成的数据包从虚拟机 B 发送到虚拟机 A，虚拟机 A 中的捕包程序会自动将其捕获，通过检查链路层和网络层以及数据的信息，证明捕获的数据包就是从虚拟机 B 此程序 libnet 中发送的数据包。

### 2. 程序结构

首先配置源目的主机的 ip 地址和 Mac 地址

初始化 libnet，调用 libnet\_init()函数，选择默认网卡 ens33（我的 linux 是这个）

想要发送 udp 数据包必须调用 libnet\_build\_udp()，libnet\_build\_ipv4()，libnet\_build\_ethernet()，构造 udp 数据报，ip 数据包，链路层帧，将源目的主机以及数据的信息写进去

调用 libnet\_write()函数发送 udp 数据报

调用 libnet\_destroy()释放资源

结束

## 四、实验结果

```
fs1170301027@ubuntu:~/networkSecurity/lab1$ sudo ./libnet
send udp successfully!
```

### Libnet 程序测试

```
fs1170301027@ubuntu:~/networkSecurity/lab1$ sudo ./capture
device: ens33
IP address: 192.168.124.0, Network Mask: 255.255.255.0
Input packet Filter: udp
datalink= 1
-----Begin Analysis-----
Packet length: 67
Source MAC : 00-0C-29-7B-7B-80==>Dest   MAC : 00-50-56-E8-86-08
Source IP : 192.168.124.130==>Dest   IP : 192.168.124.2
Protocol : IP/UNKNWN
 00 50 56 e8 86 08 00 0c 29 7b 7b 80 08 00 45 00
 00 35 30 12 00 00 0a 11 06 d1 c0 a8 7c 82 c0 a8
 7c 02 1f 90 1f 90 00 21 5f ef 54 65 73 74 20 6c
 69 62 6e 65 74 20 73 75 63 63 65 73 73 66 75 6c
 6c 79 21

T e s t   l
i b n e t   s u c c e s s f u l
l y !
```

### 实验二中捕包程序测试

```
record.txt
~/networkSecurity/lab1
保存(S)

00 12 connectivity-check 00 06 ubuntu 00 03 com

Recieve Datagram:
00 12 connectivity-check 00 06 ubuntu 00 03 com

Recieve Datagram:
00 12 connectivity-check 00 06 ubuntu 00 03 com

Recieve Datagram:

Recieve Datagram:
successfully!

Recieve Datagram:
Test libnet successfully!

Recieve Datagram:
00 1F 00 01

Recieve Datagram:
EE\FE 00 01

Recieve Datagram:
00 1F 00 81\80

Recieve Datagram:
Test libnet successfully!

Recieve Datagram:
Test libnet successfully!
```

写入文件成功

```
4 int main(){
5     libnet_t* init_libnet; // 接收初始化libnet的返回值
6     const char* device = "eth33";
7     char* err_buf;
8     //源目的主机的信息配置
9     unsigned char sourceMac[6] = {0x00,0x0C,0x29,0x7B,0x7B,0x80};
10    unsigned char targetMac[6] = {0x00,0x50,0x56,0xE8,0x86,0x08};
11    char* sourceIP = "192.168.124.130";
12    char* targetIP = "192.168.124.2";
13    //初始化libnet
14    if((init_libnet = libnet_init(LIBNET_RAW4,device,err_buf)) == NULL){
15        perror("init libnet error! \n");
16        exit(0);
17    }
18    unsigned long netSourceIP = libnet_name2addr4(init_libnet,sourceIP,LIBNET_RESOLVE); //字符串类型的IP转化为顺序网络字节流
19    unsigned long netTargetIP = libnet_name2addr4(init_libnet,targetIP,LIBNET_RESOLVE);
20
21    int lenOfPackage = 0; //数据包长度
22    char data[100]; //要发送的数据
23    lenOfPackage = sprintf(data,"%s","Test libnet successfully!\0");
24    libnet_ptag_t package;
25    //构造udp数据报
26    package = libnet_build_udp(8080,8080,8+lenOfPackage,0,(const u_int8_t*)data,lenOfPackage,init_libnet,0);
27    int id = 12306;
28    //构造IP数据包
29    package = libnet_build_ipv4(20+8+lenOfPackage,0,id,0,10,17,0,netSourceIP,netTargetIP,NULL,0,init_libnet,0);
30    //构造链路层数据帧
31    package = libnet_build_ethernet((u_int8_t *)targetMac,(u_int8_t *)sourceMac,ETHERTYPE_IP,NULL,0,init_libnet,0);
32
33    int res = 0;
34    //发送数据包
35    if((res = libnet_write(init_libnet))!=-1){
36        perror("libnet write error!\n");
37        exit(0);
38    }
39    libnet_destroy(init_libnet); // 销毁资源
40
41    printf("send udp successfully!\n");
42    return 0;
43 }
```

具体程序

## 五、心得体会（出现问题分析）

走了一遍传输层，网络层，链路层数据报的构造过程，印象很深刻。多的问题是在数据格式上的，对各层数据报的结构掌握不数量，而且还特意先运行捕包程序找了一组 ip and mac。